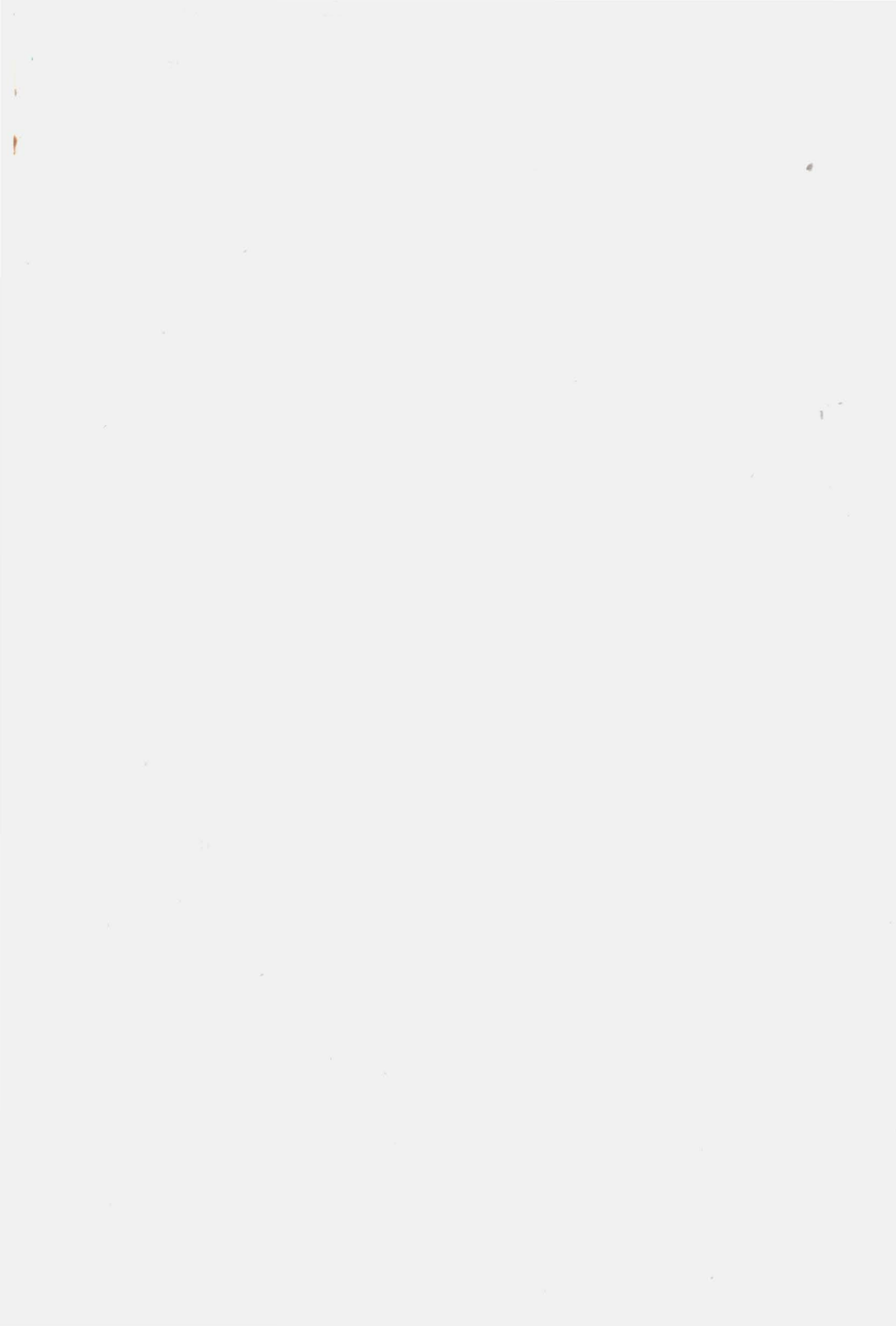
20 VIDEOLEZIONI DI BASIC
PER IMPARARE COL VIC 20

GRUPPO
EDITORIALE
JACKSON

- La tavoletta grafica
- La penna ottica
- CAD/CAM
- Alta risoluzione
- Come tracciare un punto e una retta
- Animazione e movimento di un carattere
- Videogioco n° 12

12

COMMODORE
VIC 20

VIDEO BASIC VIC 20
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea - Via Indipendenza 88 - Como
Redazione software:
Francesco Franceschini, Enrico Braglia,
Fabio Calanca
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

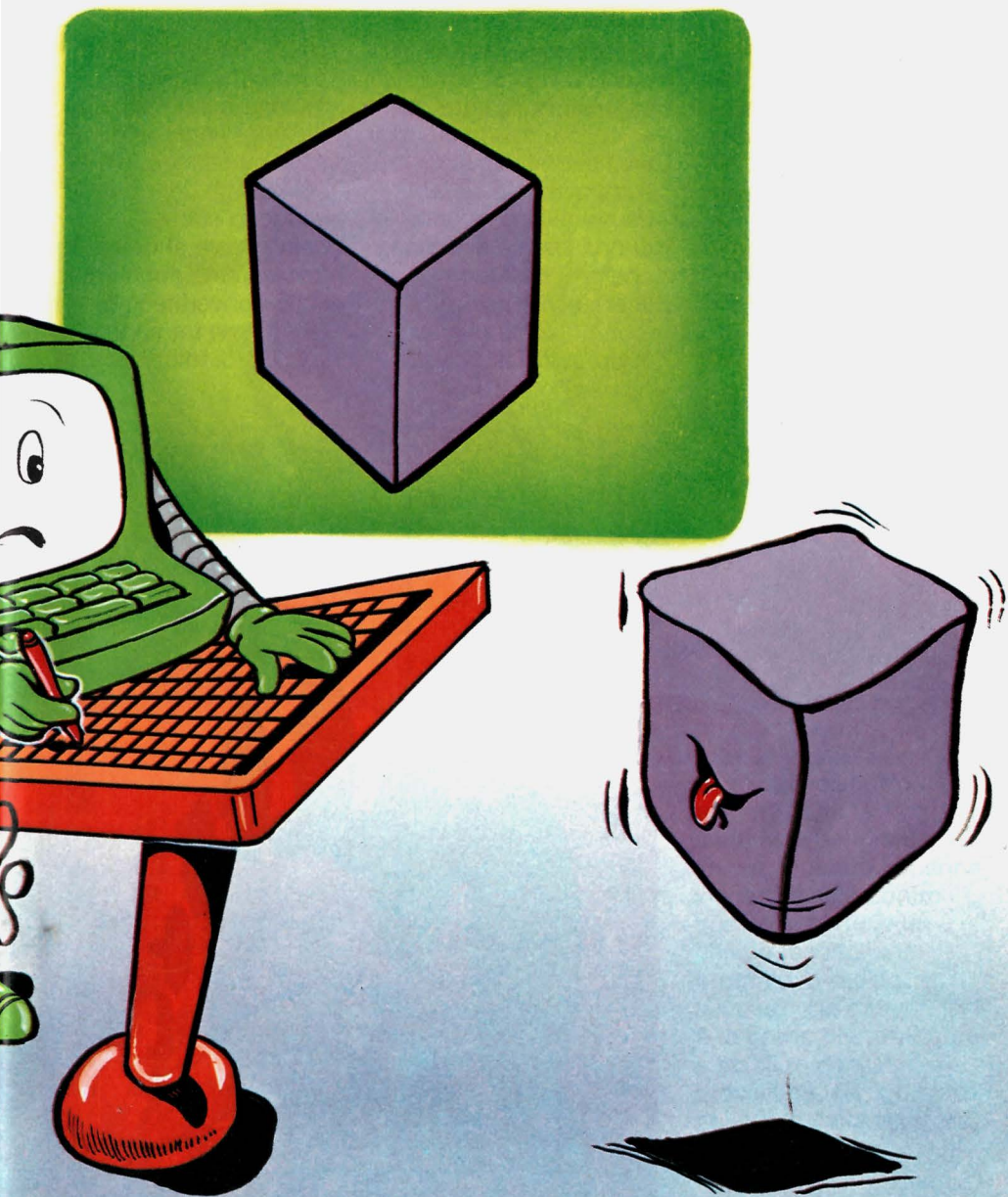
Gruppo Editoriale Jackson

SOMMARIO

Introduzione

L'elaborazione delle immagini è uno degli aspetti più spettacolari ed affascinanti di un computer. Tutti abbiamo presente le immagini trasmesse dalla televisione di sigle e disegni sviluppati con un calcolatore grafico o di progetti ed esplosi di automobili ottenuti con sistemi elettronici dedicati. Sono processi che necessitano di molta memoria e di velocità di elaborazione elevatissime e quindi quasi assenti nelle macchine di qualche anno fà.
Non oggi, però. Lo dimostra il fatto che la lotta commerciale tra le case produttrici si svolga anche "a colpi" di risoluzione grafica, di pixel indirizzabili di software più o meno capace di produrre, gestire, animare e muovere le immagini.
Entriamo in questo nuovo ambiente esaminando alcune periferiche dedicate: la penna ottica e la tavoletta grafica.

La tavoletta grafica

Sin dall'antichità il disegno si è posto come uno dei mezzi di espressione e comunicazione tra i più utili, significativi e immediati.
Era chiaro che il computer, non poteva trascurare la grafica.
Dopo i primi passi, timidi ed incerti, negli anni passati, i possibili utilizzi degli elaboratori elettronici in questo settore stanno infatti diventando sempre più numerosi ed importanti, tanto che non sembra difficile pronosticare un futuro, non troppo lontano, fatto di immagini ed animazioni totalmente gestite o composte con l'ausilio del calcolatore.
I primi esempi sono già nei cinematografi.
Si parla poi (ma non è fantascienza) della possibilità di realizzare nuovi film con interpreti scomparsi da anni e "simulati da computer".
Ovviamente, in questi casi si tratta di computer da milioni di dollari; tutti i personal computer comunque hanno notevoli capacità grafiche che li rendono utili - per non dire indispensabili - in un ampio campo di applicazioni: dalla progettazione all'urbanistica, dalla medicina agli affari, dalla didattica ai giochi. Pur essendo una macchina essenzialmente numerica, il computer è quindi in grado di comporre disegni e grafici. Naturalmente, affinché le immagini

HARDWARE

possano essere trattate, sono necessarie due cose: una codifica delle immagini stesse sotto forma di numeri e delle periferiche in grado di comunicare al computer queste immagini.
La tavoletta grafica è appunto un dispositivo periferico che consente di realizzare disegni in modo semplice ed immediato. Si tratta, in sostanza, di una piccola tavola (di dimensioni estremamente variabili da modello a modello) e di una specie di penna, che può essere liberamente posizionata (con notevole precisione) in un qualsiasi punto di questa tavola.
Una volta selezionato il punto che si desidera includere nel disegno basta premere la penna sulla tavoletta e le coordinate del punto vengono subito comunicate al computer (ovviamente dopo essere state convertite in forma digitale da un apposito circuito di interfaccia). Il

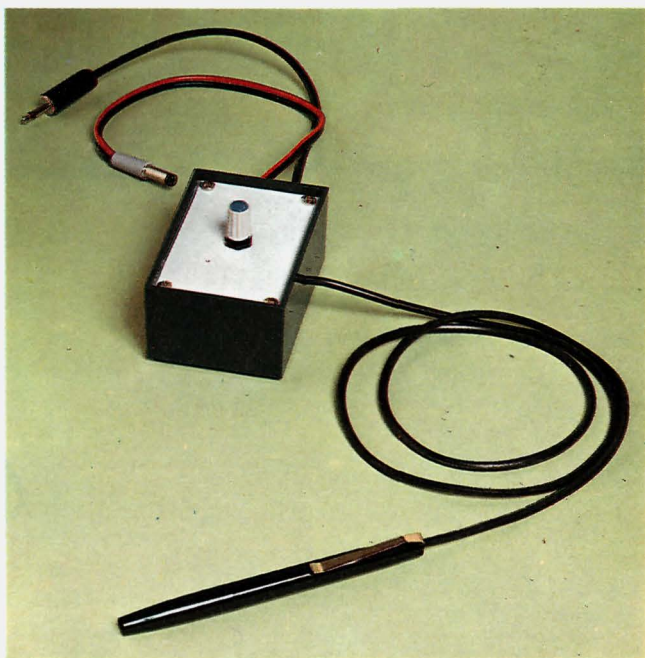
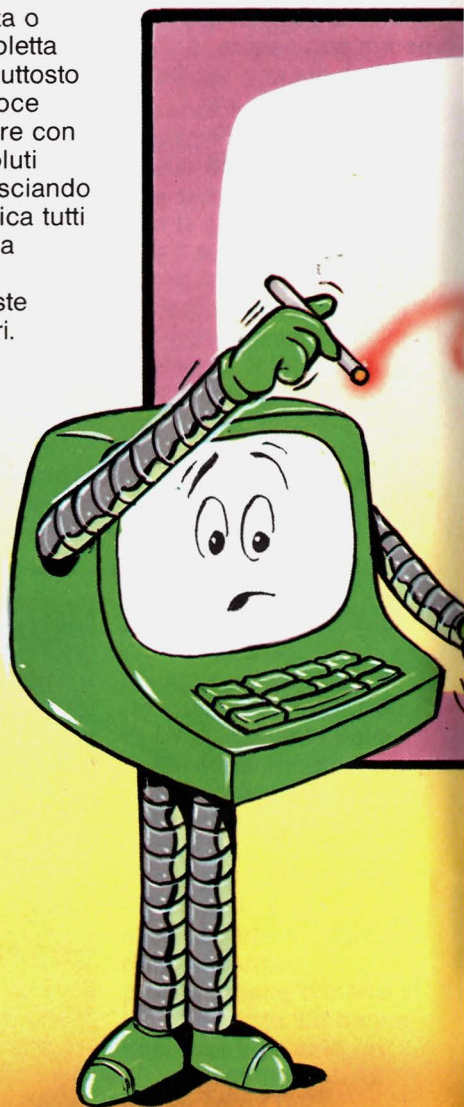
HARDWARE

complesso delle coordinate, che vengono interessate durante il movimento della penna, compone via via il disegno, che appare così sullo schermo proprio come accadrebbe se si passasse una matita sopra un comune foglio di carta.
Il principio di funzionamento che consente di eseguire queste operazioni è normalmente basato su un fitto reticolo di fili incrociati, disposto appena sotto la superficie del piano di lavoro. Quando la penna viene premuta contro questa superficie un campo magnetico, prodotto in rapida successione dai vari fili, è in grado di individuare e stabilire con precisione le coodinate del punto sulla tavoletta.
È ovvio che la risoluzione (cioè la capacità di distinguere il più piccolo movimento della penna) dipende dal

numero di fili che compongono il reticolo: nei modelli per uso hobbystico essa è logicamente molto più limitata rispetto agli strumenti professionali, arrivando comunque, anche nei modelli più economici, alle 10 linee per millimetro (la tavoletta riesce cioè ad avvertire variazioni nella posizione della penna fino a valori superiori di 1/10 di millimetro).
Come qualunque altra periferica, la tavoletta grafica serve comunque soltanto per comunicare informazioni al computer, o meglio al programma, che decide come utilizzare le informazioni ricevute.
Insieme alla tavoletta deve quindi essere utilizzato anche un programma che converta i comandi impartiti con la penna in precise istruzioni eseguibili dal computer. Per i personal più diffusi questi programmi vengono forniti assieme alla tavoletta, e costituiscono molto spesso parte integrante di tutto il corredo necessario per disegnare con il calcolatore.

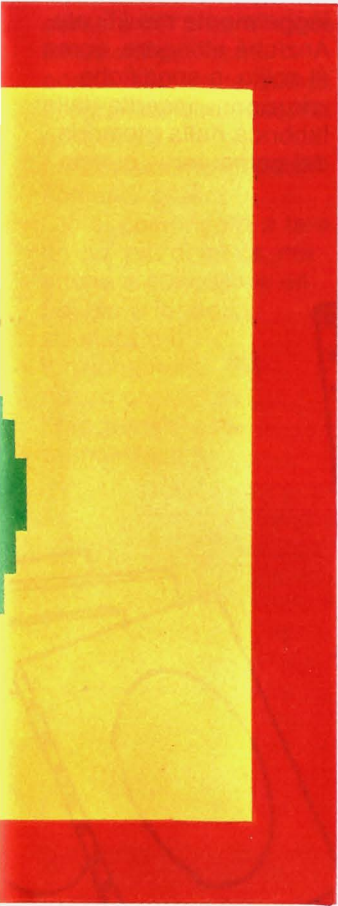
La penna ottica

Un altro dispositivo che consente di disegnare in modo più economico e meno ingombrante della tavoletta grafica è la cosiddetta penna ottica (o penna luminosa).
Si tratta di un sensore, di forma molto simile ad una comune penna, in grado di rivelare le variazioni di luminosità alle quali viene sottoposto. Il suo impiego è molto simile a quello della penna di una tavoletta grafica, con la sola differenza che il piano di lavoro è costituito - anziché da un supporto esterno - dal video stesso del computer. Infatti, è sufficiente appoggiare la penna in un qualsiasi punto del video ed il computer riceve (come al solito convertite in forma digitale) le coordinate del punto selezionato.
Per capire il modo in cui ciò possa avvenire bisogna richiamarsi per un momento al principio di funzionamento delle unità video. Ricordi? Un raggio (o fascio) di elettroni scandisce in continuazione uno strato di sostanze sensibili alla luce (i fosfori), poste sulla superficie interna dello schermo, provocandone o meno la luminescenza. La penna luminosa è proprio un dispositivo in grado di distinguere i punti illuminati da quelli non illuminati da tale raggio. Dato che la posizione del fascio elettronico è

nota in ogni istante, anche la posizione della penna può essere determinata con facilità dai circuiti adibiti al controllo del video. La precisione della penna ottica risulta comunque notevolmente più limitata di una tavoletta grafica, non potendo, in nessun caso, superare la definizione propria dello schermo. Inoltre essa costringe a lavorare con il braccio alzato e con gli occhi sempre fissi sul video. Tuttavia, non tutte le penne ottiche sono uguali. Differenze più o meno piccole nelle dimensioni e nella luminosità richiesta allo schermo video ne rendono infatti alcune migliori delle altre. È naturale che le migliori siano quelle con le dimensioni più piccole e che richiedono una minore luminosità allo schermo, visto che affaticano di meno sia il braccio che la vista. Ancora una volta, comunque, risultano indispensabili per il funzionamento di questo dispositivo i circuiti elettronici di interfaccia ed il programma di utilizzo impiegati, per rendere compatibili al calcolatore le

informazioni ricevute dall'esterno.
Ad ogni modo, per lavori di una certa complessità e precisione la penna ottica non può certo essere paragonata o sostituita alla tavoletta grafica. Essa è piuttosto un comodo e veloce mezzo per indicare con rapidità i punti voluti sullo schermo, lasciando alla tavoletta grafica tutti i lavori nei quali la complessità e la precisione richieste risultano superiori.

HARDWARE

CAD/CAM

Abbiamo visto che l'introduzione dei computer (ed in particolar modo dei personal) propone, oltre ai tradizionali utilizzi, anche la possibilità di produrre disegni tramite elaboratore.
Tale possibilità non è comunque limitata alle quotidiane attività di ufficio od al puro divertimento: tra le molteplici applicazioni, particolare risalto meritano infatti i recenti utilizzi della grafica computerizzata nel campo della progettazione industriale.
Questi utilizzi, inizialmente nati grazie agli studi fatti nelle grosse industrie automobilistiche, hanno visto il computer collaborare ed affiancarsi in misura sempre maggiore ai progettisti e ai disegnatori, consentendo il superamento di alcuni problemi che da sempre condizionavano in maniera molto pesante la fase di studio e progetto di qualsiasi prodotto industriale.
CAD e CAM (Computer Aided Design e Computer Aided Manufacturing, rispettivamente Progettazione assistita dal computer e Produzione assistita dal computer) sono proprio le abbreviazioni - universalmente conosciute - che indicano questo specifico utilizzo del computer.
Le applicazioni di simili tecnologie (poiché vedremo che di vere e proprie tecnologie si tratta) sono ormai numerosissime ed estremamente diffuse: automatizzare progettazione, lavorazione e costruzione ha difatti significato - e significa tuttora - una notevole riduzione ed ottimizzazione degli sforzi e soprattutto dei costi in diversi processi industriali, permettendo così la creazione di prodotti migliori a prezzi più contenuti.
L'esempio classico che illustra al meglio le possibilità attualmente raggiunte dal CAD/CAM riguarda l'industria dei circuiti integrati (o chip).
Sai già che un circuito integrato non è altro che un microscopico circuito elettrico posto in una piastrina di silicio.
Sembrerebbe quindi cosa facile poter rimpicciolire direttamente il circuito di progetto (realizzato necessariamente a

"misura d'uomo" e quindi con dimensioni tali da potervi lavorare sopra ad occhio nudo) nel prodotto finito, cioè *in un microscopico chip*. Niente di più errato. Sin dalla fase di progetto occorre infatti tener presente che bisogna ottimizzare e minimizzare sia gli spazi disponibili che gli elementi elettronici *costituenti il circuito*, così che alla fine risulti occupata la minor superficie possibile. Questo lavoro implica per forza di cose migliaia di calcoli, con i quali esaminare tutte le possibili disposizioni e combinazioni, che risultano pertanto assolutamente impossibili da affrontare *senza l'ausilio di un* computer appositamente predisposto.
Sul video del progettista l'elaboratore - grazie al programma CAD - visualizza allora le possibili disposizioni dei componenti elettronici, limitando (od addirittura evitando) la lunga e noiosa fase di disegno manuale dei vari circuiti.
Una volta risolto il progetto anche la produzione dei chip richiede l'adozione di particolari cautele ed il rispetto di determinate tolleranze alle macchine adibite alla lavorazione, cose queste che soltanto un computer programmato per il CAM è in grado di coordinare con affidabilità e sicurezza.
Alla fine del processo produttivo la tecnologia CAD/CAM è quindi in grado di offrire dei circuiti integrati con qualità e prestazioni nettamente superiori a quelle altrimenti ottenibili e soprattutto ad un prezzo di gran lunga più conveniente.
Quello che abbiamo fatto era solo un esempio: invece di chip potevano essere telai per automobile o componenti di motori a reazione, ed il discorso sarebbe stato praticamente identico.
È chiaro che l'uso di una simile tecnica per la progettazione e la produzione richiede computer di grandi dimensioni ed elevata potenza di calcolo (e pertanto di esclusiva portata delle grosse industrie). Già adesso stanno comunque cominciando ad apparire i primi programmi per personal computer: nel giro di pochi anni anche le piccole industrie o gli studi professionali dovrebbero quindi avere a propria disposizione questi utili e potenti mezzi di progettazione e produzione, permettendo così di allargare ulteriormente il già vasto repertorio di prodotti attualmente fabbricati utilizzando il CAD/CAM.

Alta risoluzione

All'interno del tuo computer esiste un circuito integrato chiamato VIC (Video Interface Controller, cioè controllore dell'interfaccia video), che tra le proprie funzioni ha quella di attingere alle informazioni contenute in certe regioni della memoria ed utilizzarle per creare dei segnali, che, opportunamente codificati, possano raggiungere il televisore od il monitor, generando delle immagini.
Il VIC è un dispositivo programmabile, cioè è in grado di funzionare in più modi grafici, a seconda dei comandi che gli giungono in ingresso.

Il primo di questi è il cosiddetto "modo caratteri": è quello disponibile al momento dell'accensione ed è anche il più semplice da usare.
In questo "modo" il VIC è programmato per utilizzare i dati nella memoria, ricavandone 23 righe di 22 caratteri ciascuna. I caratteri possono naturalmente essere sia quelli generati dalla ROM (2 set) riportati sui tasti sia quelli definiti in RAM dal programmatore.
Operare per caratteri coinvolge però un'area dello schermo di ben 64 punti. Ciò non consente di comporre disegni o grafici di una certa difficoltà, situazioni in cui il dettaglio è molto importante.
Per superare tale limitazione può essere allora utilizzato un altro modo grafico, chiamato "grafico ad alta risoluzione", in contrapposizione al precedente denominato a "bassa risoluzione".
L'alta risoluzione permette di considerare lo schermo come costituito da tanti puntini luminosi (pixel), tra loro completamente indipendenti. In alta risoluzione ciascun puntino viene rappresentato da un unico bit della memoria, potendo infatti essere soltanto acceso (bit 1) o spento (bit 0). Poiché il video del VIC 20 è composto da 23 linee di 22 caratteri - e ciascuno di essi è formato da 8 * 8 punti -, lo schermo risulta allora composto da ben 32384 punti, disposti secondo 184 righe da 176 punti.
Purtroppo, però, la Commodore non ha provvisto il VIC 20 di comandi adatti per disegnare in alta risoluzione. Per eseguire disegni, grafici o diagrammi dovremo quindi ricorrere ai comandi PEEK e POKE, andando cioè a lavorare direttamente sulle zone di memoria video che si desiderano modificare.

LINGUAGGIO

La grafica

Dal momento che (come hai appena visto) non esistono comandi grafici sul tuo VIC 20, la nostra lezione sul linguaggio risulta per forza di cose leggermente modificata. Anziché attingere, come al solito, a specifiche istruzioni - inserite dalla fabbrica nella memoria del computer -, questa

volta dovremo infatti cercare di "creare" delle istruzioni per conto nostro, scrivendo quindi programmi che possano simulare il funzionamento di detti comandi grafici.
Ciò ci costringerà a fare un po' più di fatica, ma anche a conoscere un po' più a fondo la struttura e il funzionamento del nostro computer.
I programmi che prenderemo in considerazione raggiungeranno gradualmente i seguenti obbiettivi:
1) passare in alta risoluzione, riservando una parte della memoria per la grafica;
2) tracciare un punto;
3) tracciare una linea retta tra due punti desiderati.
Alcune istruzioni di questi programmi potranno non risultarti completamente chiare, richiedendo per la loro spiegazione la conoscenza di nozioni che sono al di fuori dei limiti del nostro corso e che quindi non possono essere eccessivamente approfondite.
Prima di avventurarci in questa operazione sarà però bene riprendere e studiare a fondo per un istante il funzionamento degli operatori logici AND e OR, argomento questo che tu già conosci da qualche tempo, ma che oggi diventa veramente importante e fondamentale.
Vediamo per primo AND: esso opera su due numeri binari e restituisce valore 1, nel caso in cui i due numeri siano entrambi 1, e valore 0, se uno dei due o entrambi sono 0:

1 AND 1 = 1	1 AND 0 = 0
0 AND 1 = 0	0 AND 0 = 0

Se i numeri sono composti da più cifre, l'operazione viene eseguita per coppie di numeri corrispondenti:

1101 AND 1011

si esegue così

```
 1101
 1011
 ----
 1001
```

Quando in un programma l'operatore AND riceve dei numeri decimali, dopo averli trasformati nei corrispondenti valori binari esegue la AND logica e ritrasforma il risultato in numero decimale. Quindi l'istruzione:

```
PRINT 10 AND 7
```

```
10 decimale ----------> 1010 binario
 7 decimale ----------> 0111 binario

10 AND 7------> 0010 binario------> 2 decimale
```

provocherà la stampa sullo schermo del valore 2.

Eseguire l'AND di due numeri può essere utile per conoscere il valore dei bit di un byte o per azzerare un dato bit. Così, se per esempio volessimo conoscere il contenuto del quinto bit di una certa locazione di memoria, supponiamo la 3543, potremmo infatti eseguire l'AND di quel byte con il numero binario 10000 (16 decimale), cioè eseguire PRINT PEEK (3543) AND 16, ed il risultato sarebbe 10000 (ossia 16) solo se il quinto bit contenesse 1, viceversa sarebbe 0.
Anche OR opera su due numeri, trasformandoli in binari e generando quindi su ogni coppia di numeri corrispondenti il valore 0, se entrambi i numeri sono nulli, ed il valore 1 negli altri casi.

```
0 OR 0 = 0        0 OR 1 = 1
1 OR 0 = 1        1 OR 1 = 1
```

```
12  OR  6
```

viene quindi eseguito in questo modo:

```
12 decimale ------>  1100 binario
 6 decimale ------>  0110 binario

12 OR 6 --------> 1110 binario --------> 14 decimale
```

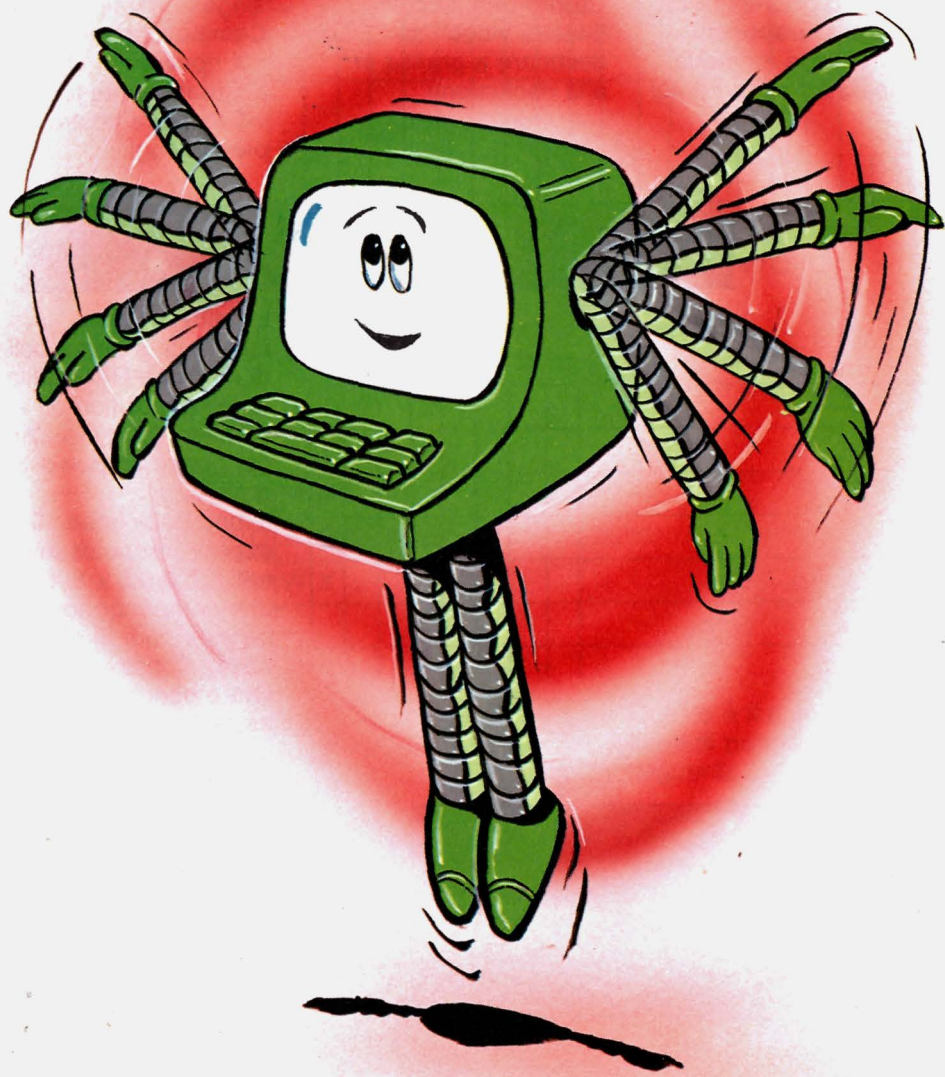
Gli operatori AND e OR servono, usando una POKE a modificare una parte di un byte letto con una PEEK settando o resettando determinati bit. Per esempio, dato il byte 10011101 (decimale 157), per azzerare il terzo e il quarto bit si userà il numero 11110011 (decimale 243) e l'operatore logico AND: infatti 157 AND 243 = 145, ossia 10010001. A questo punto con POKE si inserirà tale valore nella cella di memoria letta prima con PEEK e si sarà giunti al risultato desiderato, cioè alla modifica dei due bit.

Come entrare in alta risoluzione

Per entrare con il tuo Commodore in modo alta risoluzione è necessario intervenire sul chip 65161 (VIC) il quale controlla direttamente tre aree di memoria:

1) la memoria di schermo (MAPPA VIDEO),
2) la memoria colore (MAPPA COLORI),
3) la memoria caratteri (MAPPA CARATTERI).

La MAPPA VIDEO è una zona di 506 byte nella RAM; ogni byte corrisponde a una delle posizioni dell'immagine sul video; il primo byte corrisponde alla prima posizione in alto a sinistra, il secondo alla posizione successiva sulla stessa linea e così via.
Ricorda che le dimensioni del quadro video sono di 23 righe di 22 caratteri, per un totale appunto di 506 caratteri.
Ogni byte contiene il codice del carattere (non in ASCII, ma in DISPLAY CODE, che deve comparire nella posizione corrispondente; esso fa da puntatore ad una zona della mappa caratteri in cui è definito il carattere stesso.
La MAPPA VIDEO inizia alla locazione 7680 (1E00H) nella configurazione standard del VIC 20 e in quella con 3K di espansione, mentre viene spostata alla locazione 4096 (1000H) nella versione con espansioni di memoria superiori a 3K.
La posizione della mappa video è controllata da alcuni bit dei registri 2 e 6 del 6561, come si vedrà più avanti; inoltre, al momento dell'accensione del calcolatore il byte 648 contiene 30 o 16, che, moltiplicati rispettivamente per 256, danno l'indirizzo di inizio di questa mappa (30 * 256 = 7680, 16 * 256 = 4096).
La MAPPA COLORI è una zona di 506 byte nella RAM, ognuno dei quali corrisponde, anche in questo caso, a una posizione dello schermo, con lo stesso criterio visto per la mappa video.
I 3 bit meno significativi di ogni byte (bit 0-2) contengono il codice di uno tra 8 colori disponibili; il bit 3 seleziona il modo "alta risoluzione" quando è a

0 (e questa è la norma), e il modo "multicolor" quando è a 1. Nel primo caso il colore indicato dai bit 0-2 viene assegnato al carattere o al suo sfondo, in dipendenza dello stato del bit 3 del registro 16, di indirizzo 36879 (900FH), del 6561.
La mappa colori inizia alla locazione 38400 (9600H) nella configurazione standard del VIC e in quella con 3K di espansione, mentre viene spostata alla locazione 37888 (9400H) nelle altre configurazioni.
La MAPPA CARATTERI è la zona di memoria in cui sono descritti, punto per punto, i diversi caratteri da visualizzare. Il set standard di caratteri del VIC 20 è definito in ROM a partire dalla locazione 32768 (8000H), ma, come già sai, puoi definire in RAM una nuova mappa caratteri, in base alle tue esigenze.
Puoi infatti modificare l'indirizzo di partenza della mappa caratteri, tramite il registro 6 (di indirizzo 36869); questo consente di costruire in RAM una mappa caratteri personalizzata e di istruire il calcolatore ad usarla in alternativa alla mappa standard.
Nelle applicazioni di grafica ad alta risoluzione puoi immaginare lo schermo come una finestra composta da un certo numero di caratteri grafici, di volta in volta definiti in base alle esigenze del tipo richiesto di elaborazione.
Ora, poiché l'intero video è composto da 22 x 23 = 506 caratteri, ognuno dei quali richiede 8 byte per essere definito (in totale 4048 byte), sarebbero necessari ben 4K di memoria RAM per gestire tutto lo schermo in alta risoluzione.
I 3,5K RAM della configurazione base del VIC 20 non lo consentono: per questa ragione in alta risoluzione si usano porzioni di schermo, calibrate di volta in volta per le particolari necessità, tenuto conto anche che una parte della RAM disponibile deve essere riservata al programma BASIC.

LINGUAGGIO

Inizializzazione dello schermo

Come un buon disegnatore predispone con cura il supporto sul quale tracciare il proprio lavoro, anche noi ci dobbiamo premunire da possibili scarabocchi presenti nella memoria destinata ad apparire sul video.
Se, come sai, ad ogni bit a 1 corrisponde un punto acceso, per avere un foglio pulito sarà sufficiente azzerare tutti i bit dei byte che lo compongono.
Supponiamo ora di voler "aprire una piccola finestra video" da destinare a grafica in alta risoluzione.
Per semplicità definiamone larghezza ed altezza pari a 5 caratteri e posizioniamola in alto a sinistra. Avremo così a disposizione 40 x 40 = 1600 punti, che potremo indirizzare (accendere) singolarmente.
Cominciamo innanzitutto col ripulire accuratamente tutto il video, poi il nostro piccolo foglio.

```
5 PRINT "♥"
10 FOR I = 7168 TO 7168 + 25 * 8 : POKE I, 0 : NEXT I
```

L'indirizzo 7168 individua la prima cella di memoria RAM nella quale depositiamo i caratteri (le informazioni) del nostro schermo grafico.
Il loop della linea 10 azzera scrupolosamente tutti i 200 (25 x 8) byte che compongono i 25 caratteri (5 x 5) della nostra finestra.
Ora è necessario assegnare ad ognuno dei 25 caratteri un codice schermo diverso, in modo che ognuno sia noto ed indipendente dalle variazioni di un altro.
Inizieremo col primo *codice* (0), corrispondente alla @, tenendo conto però che, diversamente dallo

LINGUAGGIO

schermo normale, il carattere E non si troverà sulla prima riga (N° 0) in sesta colonna (N° 5), ma, come risulta più chiaro dalla tabella riportata, nella prima colonna (N° 0) della seconda riga (N° 1).

NELLA MAPPA CARATTERI SONO ORGANIZZATI COSI'

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	V	W	X	Y	Z																	
2																						
3																						
4																						
5																						
6																						
7																						

NEL NUOVO SCHERMO GRAFICO DA NOI DEFINITO
LI ORGANIZZIAMO COSI'

	0	1	2	3	4	5	6	7	8	9	10	11
0	@	A	B	C	D							
1	E	F	G	H	I							
2	J	K	L	M	N							
3	O	P	Q	R	S							
4	T	U	V	W	X							
5												
6												

Questo è il compito delle linee di programma seguenti:

```
20 FOR Y = 0 TO 4
30 FOR X = 0 TO 4
40 POKE 7680 + Y * 22 + X, Y * 5 + X

60 NEXT X
70 NEXT Y
```

Nel caso la linea 40 ti risulti un po' astrusa, prova ad annotare su un pezzo di carta i valori assunti nel corso dei loop da Y e da X e dalle espressioni in cui compaiono.

Codici schermo

SERIE1	SERIE2	POKE	SERIE1	SERIE2	POKE	SERIE1	SERIE2	POKE
@		0	U	u	21	*		42
A	a	1	V	v	22	+		43
B	b	2	W	w	23	,		44
C	c	3	X	x	24	—		45
D	d	4	Y	y	25	.		46
E	e	5	Z	z	26	/		47
F	f	6	[		27	Ø		48
G	g	7	£		28	1		49
H	h	8	]		29	2		50
I	i	9	↑		30	3		51
J	j	10	←		31	4		52
K	k	11	SPACE		32	5		53
L	l	12	!		33	6		54
M	m	13	"		34	7		55
N	n	14	#		35	8		56
O	o	15	$		36	9		57
P	p	16	%		37	:		58
Q	q	17	&		38	;		59
R	r	18	'		39	<		60
S	s	19	(		40	=		61
T	t	20	)		41	>		62

SERIE1	SERIE2	POKE	SERIE1	SERIE2	POKE	SERIE1	SERIE2	POKE
?		63		T	84			106
		64		U	85			107
	A	65		V	86			108
	B	66		W	87			109
	C	67		X	88			110
	D	68		Y	89			111
	E	69		Z	90			112
	F	70			91			113
	G	71			92			114
	H	72			93			115
	I	73	π		94			116
	J	74			95			117
	K	75	SPACE		96			118
	L	76			97			119
	M	77			98			120
	N	78			99			121
	O	79			100		✓	122
	P	80			101			123
	Q	81			102			124
	R	82			103			125
	S	83			104			126
					105			127

Tieni inoltre presente che, come visto poc'anzi, alla locazione 7680 ha inizio la MAPPA VIDEO. Ti sarai forse accorto dell'assenza della linea 50: è stata momentaneamente tralasciata per consentirti la massima concentrazione sui loop di organizzazione del piccolo schermo in alta risoluzione. Ma rimediamo subito.
Per far comparire un carattere sul video occorre scrivere sia il codice del carattere nella mappa video che il codice del colore nella mappa colori. Se dimentichi il codice colore, non si vede il carattere; infatti il sistema, al momento dell'accensione, assegna il colore bianco sia ai caratteri che allo sfondo.
Il ruolo della linea 50 consiste nel predisporre un colore di inchiostro, nel nostro caso il viola (4), per ogni carattere dello schermo appena definito.

```
50 POKE 38400 + Y * 22 + X, 4
```

La mappa colori, come già detto, inizia alla locazione 38400.
Ora non resta che attivare il nostro videografico, modificando opportunamente il puntatore alla nuova mappa caratteri:

```
80 POKE 36869, 255
```

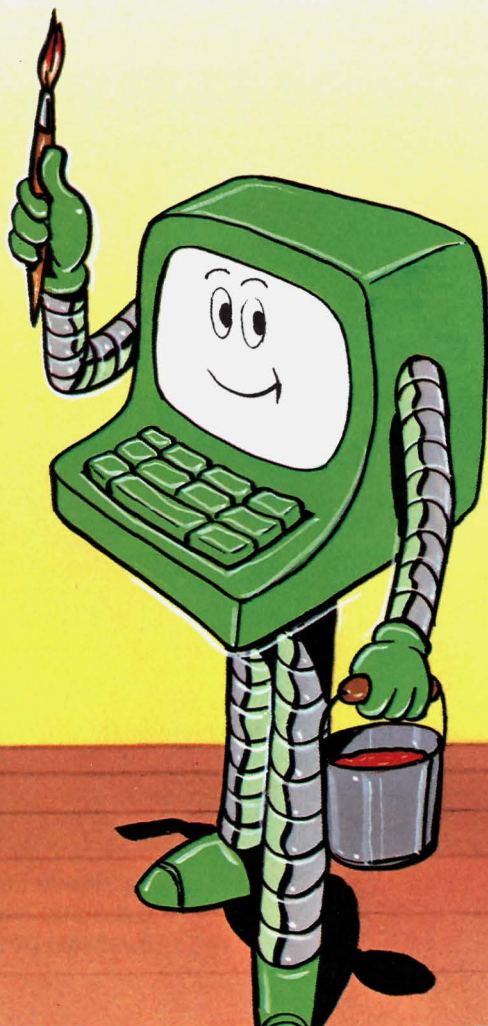
Adesso il VIC 20 guarderà alla zona di memoria RAM che inizia dalla locazione 7168 come alla nuova fonte di informazioni da proiettare sullo schermo.
Riassumendo, il programma di inizializzazione si presenta così:

```
 5 PRINT "▣"
10 FOR I = 7168 TO 7168 + 25 * 8 : POKE I, 0
   : NEXT
20 FOR Y = 0 TO 4
30 FOR X = 0 TO 4
40 POKE 7680 + Y * 22 + X, Y * 5 + X
50 POKE 38400 + Y * 22 + X, 4
60 NEXT X
70 NEXT Y
80 POKE 36869, 255
```

Puoi ora farlo girare: osservando attentamente la parte dello schermo in alto a sinistra, vedrai temporaneamente l'effetto della linea 40, poi, per effetto dei byte azzerati, una videata perfettamente pulita.
Per evitare che il READY di fine esecuzione programma (in realtà sotto forma di strani geroglifici) vada a sporcare lo schermo, aggiungi momentaneamente la seguente linea:

```
90 GOTO 90
```

che innesca un loop infinito, da interrompere con la pressione del tasto RUN/STOP.
Per riportare il VIC 20 alle condizioni normali premi contemporaneamente i tasti RUN/STOP e RESTORE.
Il programma, come premesso, si riferisce ad una finestra video di 25 (5 x 5) caratteri. Se vuoi, puoi inizializzarne una di dimensioni maggiori. Sarà sufficiente in tal caso modificare opportunamente il numero caratteri nella linea 10, i contatori di ciclo e la costante (5 nell'esempio) della linea 40.
Per inizializzare, ad esempio, uno schermo in alta risoluzione di 6 x 8 (48) caratteri occorrerà un programma del tipo:

```
 5 PRINT "♥"
10 FOR I = 7168 TO 7168 + 48 * 8 : POKE I, 0 : NEXT
20 FOR Y = 0 TO 7
30 FOR X = 0 TO 5
40 POKE 7680 + Y * 22 + X, Y * 6 + X
50 POKE 38400 + Y * 22 + X, 4
60 NEXT X
70 NEXT Y
80 POKE 36869, 255
```

Il punto

Adesso che conosciamo la via da seguire per entrare nel modo alta risoluzione, e per cancellare la pagina grafica, vogliamo provare ad accendere e disegnare uno specifico

punto tra i 1600 (25 x 64) che compongono la pagina grafica appena definita

(8 x 5 = 40 in orizzontale)
(8 x 5 = 40 in verticale).

La prima cosa che occorre fare è allora scegliere un riferimento per l'identificazione dei singoli punti (assegnando cioè una coppia di valori a ciascuno dei 1600 punti): le righe e le colonne della pagina grafica definiscono infatti un sistema di coordinate - molto simile a quello utilizzato nel gioco della battaglia navale - che ben si presta a una simile operazione.
Chiamando le coordinate X e Y (X per le colonne ed Y per le righe), poniamo l'origine del nostro sistema di riferimento (convenzionalmente viene indicato come "origine" il punto di coordinate (0, 0)) nell'angolo in alto a sinistra. Orientiamo quindi l'asse delle X da sinistra verso destra e l'asse delle Y dall'alto verso il basso; in questo modo, per esempio, l'angolo in basso a destra sarà il punto (39, 39), mentre l'angolo in alto a destra sarà il punto (39, 0).
Il nostro programma per il tracciamento di un punto utilizza proprio questo modo per identificare ogni punto dello schermo

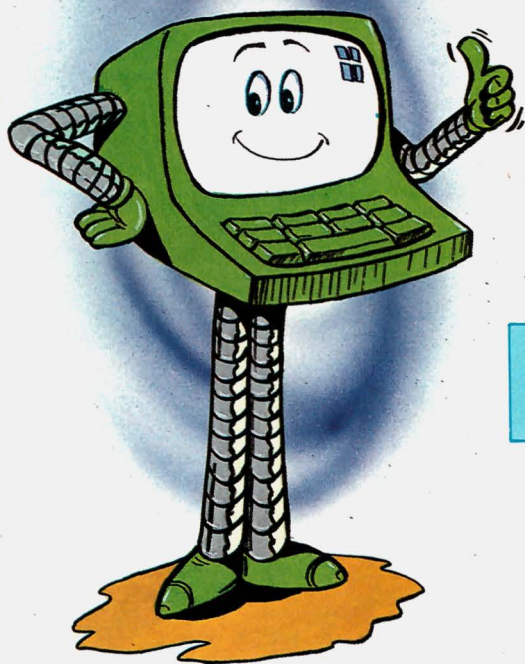
```
  5 PRINT "■"
 10 FOR I = 7168 TO 7168 + 25 * 8 : POKE I, 0 : NEXT
 20 FOR I = 0 TO 4
 30 FOR J = 0 TO 4
 40 POKE 7680 + I * 22 + J, I * 5 + J
 50 POKE 38400 + I * 22 + J, 4
 60 NEXT
 70 NEXT
 80 POKE 36869, 255
100 READ X : READ Y
110 DATA 21, 19
200 CAR = 7168 + Y + INT (X/8) * 40
210 BIT = 2 ↑ (7 − X AND 7)
220 POKE CAR, BIT OR PEEK (CAR)
300 GOTO 300
```

La chiave della stampa del punto (vale per qualsiasi punto

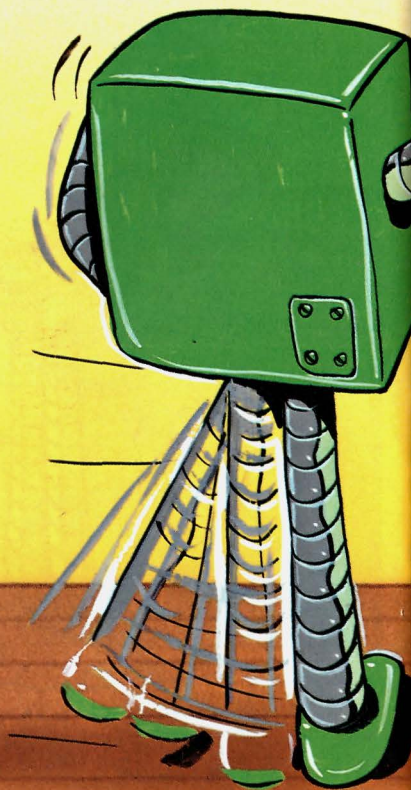
all'interno dello schermo grafico) sta nelle linee 200-210-220.
La linea 200 individua il carattere e nel carattere il byte interessato alla variazione.
La 210 seleziona il bit corrispondente al pixel da accendere.
La 220 accende il pixel.
La 300 innesca un LOOP senza fine, per mantenere intatta la visualizzazione dello schermo grafico.

Come al solito, per interrompere premi i tasti RUN STOP e RESTORE.
Per indirizzare (accendere) punti diversi modifica i valori della linea DATA (110), facendo attenzione ad attribuire alla X e alla Y valori compresi tra 0 e 39.
Ogni volta che vorrai disegnare un punto sullo schermo non avrai altro da fare che eseguire questo programma, specificando le coordinate del punto stesso.
Una possibile (anche se abbastanza inutile) applicazione del nostro programma potrebbe essere quella di far tracciare i punti in modo casuale dal tuo VIC 20.
Per poter fare ciò è sufficiente che tu sostituisca le righe 100-110 e 300 con le seguenti:

```
100 X = INT (RND (1) * 39)
110 Y = INT (RND (0) * 39)
300 GOTO 100
```

Le coordinate verranno allora impostate automaticamente dal computer, creando un ciclo infinito, che, a poco a poco, riempirà lo schermo di punti.

La retta

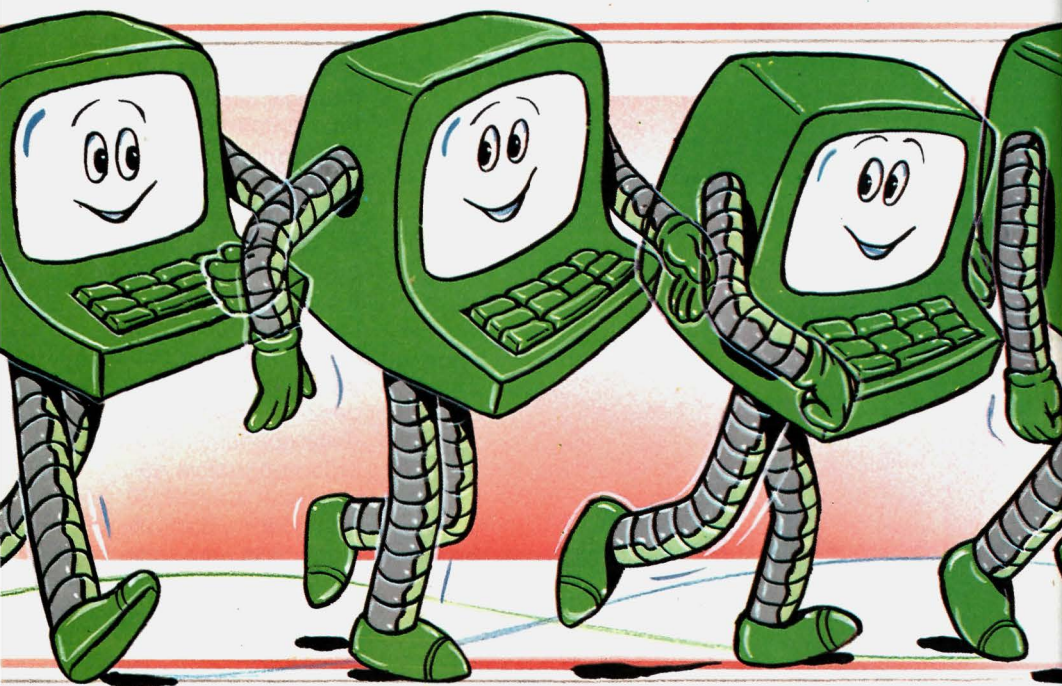
Eccoci all'ultimo programma grafico, cioè quello per tracciare una retta tra due punti qualunque. Ormai non abbiamo più particolari problemi: la nostra retta sarà infatti generata componendo tanti punti uno accanto all'altro, in modo da apparire sul video come se costituissero un'unica linea continua.

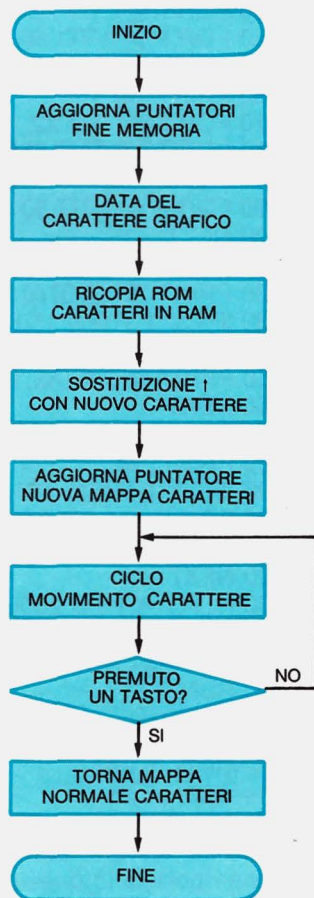
Dal momento che noi sappiamo disegnare i punti, basterà determinare i pixel da accendere (mediante una semplice espressione che tenga conto degli estremi da unire) e tutto sarà risolto in un attimo. Senza entrare in noiosi dettagli matematici eccoti direttamente il listato del programma:

```
  5 PRINT "▣"
 10 FOR I = 7168 TO 7168 + 25 * 8 : POKE I, 0 : NEXT
 20 FOR I = 0 TO 4
 30 FOR J = 0 TO 4
 40 POKE 7680 + I * 22 + J, I * 5 + J
 50 POKE 38400 + I * 22 + J, 4
 60 NEXT
 70 NEXT
 80 POKE 36869, 255
100 FOR X = 0 TO 39
110 Y = X
200 CAR = 7168 + Y + INT (X/8) * 40
210 BIT = 2 ↑ (7 - X AND 7)
220 POKE CAR, BIT OR PEEK (CAR)
230 NEXT X
300 GOTO 300
```

Come esercizio (senza andare troppo sul difficile), prova ad introdurre delle istruzioni che provochino - in maniera simile a quanto fatto prima con il punto - il disegno di rette generate casualmente dal computer.
Ad ogni modo, se in questa lezione qualcosa ti è sfuggito o ti è risultato poco chiaro, non te ne preoccupare più di tanto. Come già detto, la grafica è una delle applicazioni tra le più impegnative da utilizzare sul VIC 20, richiedendo approfonditi studi della macchina, grande pazienza e notevole abilità di programmazione.

Movimento di un carattere

L'obiettivo di questo programma è quello di muovere sullo schermo un carattere grafico da noi definito.
Al posto dell'istruzione PRINT utilizzeremo la POKE all'interno di un ciclo, per stampare direttamente nella memoria video il carattere, in modo da dare l'illusione del movimento.

0	0	1	1	1	1	0	0	60
0	1	1	1	1	1	1	0	126
1	1	0	1	1	0	1	1	219
1	1	1	1	1	1	1	1	255
1	1	0	0	0	0	1	1	195
0	1	1	1	1	1	1	0	126
0	1	0	1	1	0	1	0	90
1	1	0	0	0	0	1	1	195

```
10 POKE 51, 0 : POKE 52, 28 : POKE 55, 0 : POKE 56, 28
15 DATA 60, 126, 219, 255, 195, 126, 90, 195
20 LET M 1 = 7168 : LET M 2 = 32768
25 FOR C = 0 TO 511
30 POKE M 1 + C, PEEK (M 2 + C) : NEXT C
35 FOR C = 7408 TO 7415
40 READ A : POKE C, A : NEXT C
45 PRINT "[♥][Q][■] ↑ [■]MOVIMENTO SPRITE[■] ↑ [■]"
50 POKE 36869, 255
55 LET M 1 = 7790 : LET M 2 = 38510
60 FOR C = 0 TO 21
65 POKE M 1 + C, 30 : POKE M 2 + C, 6
70 FOR D = 1 TO 90 : NEXT D
75 POKE M 1 + C, 32 : NEXT C
80 GET A$ : IF A$ = "  " THEN GOTO 50
85 POKE 36869, 240 : END
```

Alla linea 10 la RAMTOP viene modificata, così che il BASIC non possa sconfinare nella nuova mappa caratteri situata tra gli indirizzi 7168 e 7679.
Il ciclo delle linee 25 e 30 ricopia nella nuova mappa i primi 64 caratteri direttamente dalla ROM (32768).
Il LOOP in linea 35, 40 modifica i byte che compongono il carattere ↑ con quello codificato nella linea DATA (15).
Dopo aver aggiornato nella linea 50 il puntatore, affinché il VIC 20 cerchi la forma dei caratteri da stampare nella nuova mappa caratteri, le istruzioni comprese nel ciclo FOR NEXT tra le linee 60 e 75 provvedono al movimento del carattere.

VIDEOESERCIZI

```
10 A$ = "VIDEOBASIC JACKSON"
20 PRINT "♥"
30 FOR I = 1 TO 22 : PRINT MID$ (A$, I, 1);
40 FOR P = 1 TO 50 : NEXT
50 NEXT
60 PRINT "S"
70 FOR I = 1 TO 22
80 PRINT "S ]]]]] ]]]]]]]]]]]]]]]]]" MID$
   (A$, I, 1);
90 FOR P = 1 TO 50 : NEXT
100 NEXT
110 FOR I = 22 TO 1 STEP - 1
120 PRINT "S QQQ QQQQQQQQQQQQQQQQQQQ"
130 IF I = 1 THEN 150
140 FOR S = 1 TO I - 1 : PRINT "Q"; : NEXT
150 PRINT MID $ (A$, I, 1);
160 NEXT
170 FOR I = 22 TO 1 STEP - 1
180 PRINT "S"
190 IF I = 1 THEN 210
200 FOR S = 1 TO I - 1 : PRINT "Q"; : NEXT
210 PRINT MID$ (A$, I, 1);
220 NEXT
230 FOR I = 1 TO 2000 : NEXT
240 PRINT "S QQQQQQQQQQ] PREMI UN TASTO"
250 GET A$ : IF A$ = "   " THEN 250
260 RUN
```

La vivacità di una schermata in un programma dipende anche da "trovate" come quella riportata in questo esercizio. È un'idea come altre che tu, su questa base, puoi ora sviluppare.

SPONSOR UFFICIALE
FORNITORE UFFICIALE
Azzurra
SFIDA ITALIANA AMERICA'S CUP 1987
RSCG
"Azzurra, non senti come mi batte il cuore?"
In certi casi anche un computer fa i conti con i propri sentimenti, non rimane a guardare freddo e impassibile. Per prepararci alla sfida australiana, seguiamo le prove e gli allenamenti durissimi di Azzurra. Memorizziamo ed elaboriamo i dati digitali ricavati dalle informazioni analogiche sul comportamento della barca in mare. È così che ci sforziamo di rendere ottimali le prestazioni di Azzurra in gara. Honeywell svolge il proprio lavoro come un membro dell'equipaggio: con scrupolo, passione e tanta emozione.
Conoscere e risolvere insieme.
Honeywell
Honeywell Information Systems Italia